Attempting to Understand the Diffusion of Collaboration Technology

Rick Stevens

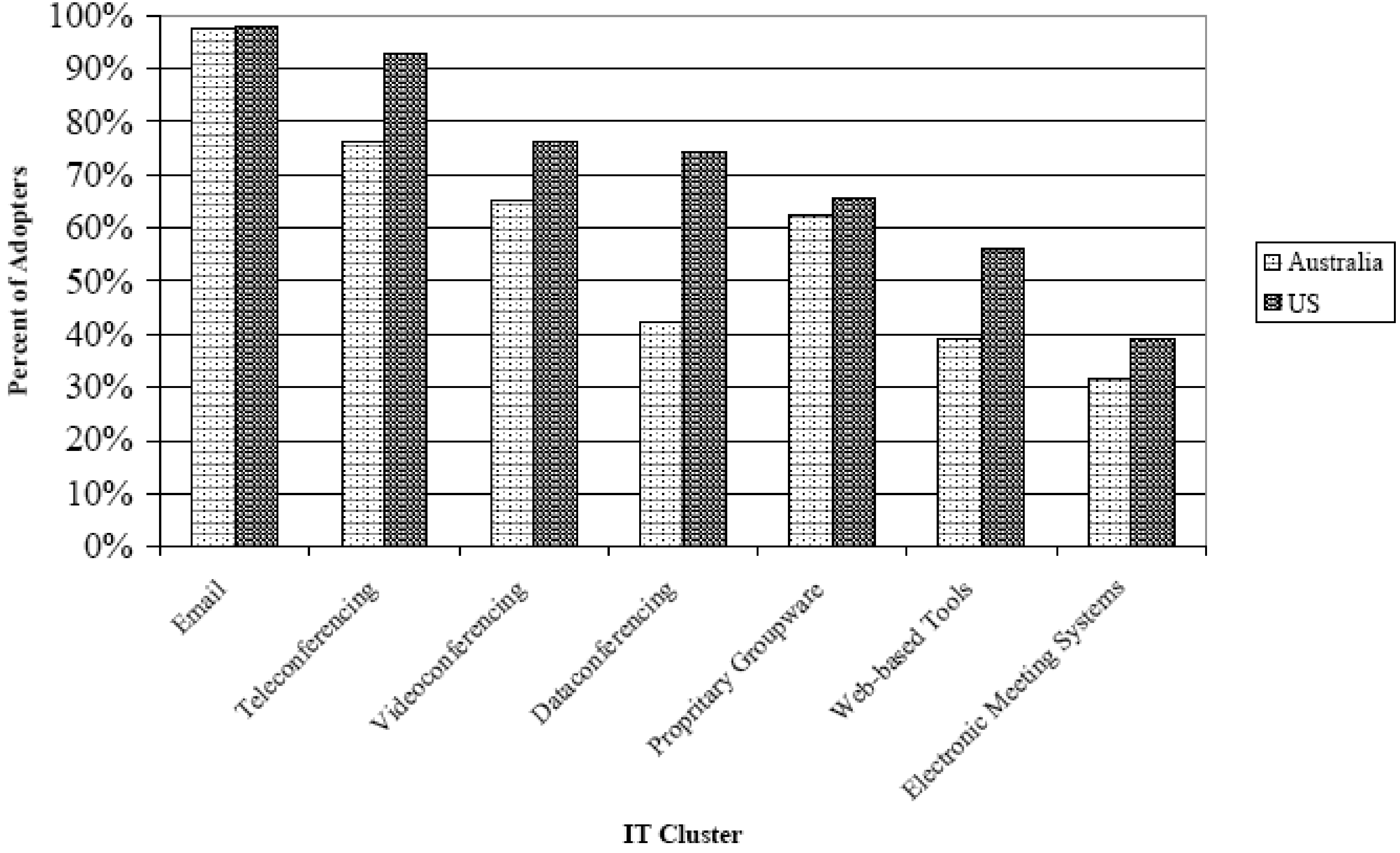
Argonne National Laboratory

The University of Chicago

There is nothing more difficult to plan, more doubtful of success, nor more dangerous to manage than the creation of a new order of things… Whenever his enemies have the ability to attack the innovator they do so with the passion of partisans, while others defend him sluggishly, so that the innovator and his party alike are vulnerable.

—Niccolò Machiavelli, *The Prince*

**Figure 5: Adoption of IT for Collaboration Support
[Australia vs. US]**



Typical Technology Development Trajectory

- Idea (1 - 3 years) [failure rate is (and should be) high]
  - Most ideas are bad (not improvements on the status quo)
  - It can take several years to have a good one
- Prototype (.1 - 2 years) [successful prototypes ~ 1 in 3]
  - Until the idea is manifest in software it can’t really be evaluated
  - The rate of producing a prototype is often rate limiting
- Product (1-2 years) [successful software product ~ 1 in 6]
  - The product needs to fix the conceptual and implementation bugs in the prototype
  - Product needs to take into account user feedback
- Infrastructure (2 - 10 years) [perhaps 1 in 5 products become infrastructure]
  - Product is integrated with the normal working environment and supported
  - Broad usage is assumed in the support structure for the organization

Two Good Sources (Moore and Rogers)

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

A Common Model of Technology Adoption

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

“The Chasm”

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

I = Innovators (folks that love technology for technology sake)

EA = Early Adopters (believe in vision to adopt technology)

C = Chasm .. Time GAP Between EA and Pragmatists

P = Pragmatists.. Risk adverse but want advantages

T = Traditionalists (Laggards) do not engage with HT products except to block them

Early Adopters

- Technology focused
- Proponents of revolutionary change
- Visionary users
- Project oriented
- Willing to take risks
- Willing to experiment
- Individually self-sufficient
- Tend to communicate horizontally (focused across disciplines)

Early Majority

- Not technology focused
- Proponents of evolutionary change
- Pragmatic users
- Process oriented
- Averse to taking risks
- Look for proven applications
- May require support
- Tend to communicate vertically (focused within a discipline)

One Person’s data

(from Michael Radwin’s BLOG)

- email - 1992 (innovator)
- web browsing - 1994 (innovator)
- Linux - 1994 (innovator)
- web publishing - 1995 (early adopter)
- e-commerce - 1996 (early adopter)
- PDA - 1997 (early adopter)
- vanity email address - 1998 (early adopter)
- cell phone - 1998 (early majority)
- snowboarding - 1998 (early majority)
- MP3 ripping - 1999 (early adopter)
- electric toothbrush - 2002 (late majority)
- broadband Internet - 2000 (early majority)
- P2P file sharing - 2000 (early majority)
- TiVo - 1999 (early adopter)
- LCD projector TV - 2001 (early adopter)
- DVD player - 2001 (early majority)
- 802.11b wireless network - 2002 (early majority)
- CD burner - 2002 (late majority)
- blog - 2002 (early adopter)
- antilock brakes - 2002 (late majority)
- digital camera - maybe this year (early majority?)
- hybrid or electric car - 2005? (early majority?)
- Lasik eye surgery - never (laggard)
- digital watch - never (laggard)

The early majority share some of the early adopter's ability to relate to technology, but ultimately they are driven by a strong sense of practicality. They know that many of these newfangled inventions end up as passing fads, so they are content to wait and see how other people are making out before they buy in themselves.

www.radwin.org/michael/blog/2003/02/standing_on_both_sides_of_.html

Too Much Technology

QuickTime™ and a
TIFF (Uncompressed) decompressor
are needed to see this picture.

Late adopters (majority) are often more interested in user experience than the details of the technology

This represents a challenge to developers!!

The Must Have Value Proposition:

A Successful Collaboration Technology Must (one or more):

1. Create a previously unavailable capability that makes collaboration dramatically easier, or more productive, or more enjoyable.

2. Visibly, verifiably and significantly reduce current overall operating costs.

3. Radically improve the productivity of the research team on a critical success factor that is already well understood.

A Case Study

- Water Boiling in Peru

What Can be Done to Improve our Odds?

- Finding (creating) right kind of partnerships
  - Open minded end users
    - Willing to try stuff that might not be obvious
  - Pragmatic Computer Scientists/Developers
    - Willing to build stuff that might seem mundane
- Dumping bad ideas and replacing them with good ones is appropriate and should be done early and often
  - don’t waste time once it is clear something is a failure (higher number of failures per second is good)
- Recognize and celebrate successes
  - True wins are rare and it is really hard to move stuff from prototype to product and from product to infrastructure
  - 10-15 years is the right timeframe for evaluating the impact of a particular idea

Steps We Need to Take to Improve the Technology Q for our Users

- Invest in Education and Training
  - Sponsor this effort directly or via Centers
- Produce User Oriented Publications
  - What’s New is a helpful way to diffuse knowledge
- Free Lunch Technology Workshops
  - Development groups should gently evangelize
- Provide Accessible Web Pages
  - Not just for developers
- Publish (Neighbor) Statistics
  - Gently shame users into adopting the latest
- Conduct High Visibility Demonstrations
  - Create Buzz but avoid HYPE

What we should be doing that we are not doing!!

- DOE (and NSF etc.) should be studying how the technology that exists is being used and how new technology is being adopted.
  - The CS research community generally lacks insightful feedback on the use of tools
  - CS community is largely unaware of the social issues that impact the rate of technology adoption
- CS peer review is not likely to get to core tools needed by applications
- User peer review is not going to push the envelope in a way that CS developers would like
- Research priorities should be established jointly between developers and end users